```
EEEEEEEEEEEEEEEE  DDDDDDDDDDD     TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEE  DDDDDDDDDDD    TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE  DDDDDDDDDDD   TTTTTTTTTTTTTTTT
EEE                 DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEEEEEEEEEEE        DDD     DDD          TTT
EEEEEEEEEEEE        DDD     DDD          TTT
EEEEEEEEEEEE        DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEE                 DDD     DDD          TTT
EEEEEEEEEEEEEEEE    DDDDDDDDDDD          TTT
EEEEEEEEEEEEEEEE    DDDDDDDDDDD          TTT
EEEEEEEEEEEEEEEE    DDDDDDDDDDD          TTT
```

```
CCCCCCCC  HH      HH  MM      MM  MM      MM  EEEEEEEEEE  SSSSSSSS   SSSSSSSS
CCCCCCCC  HH      HH  MM      MM  MM      MM  EEEEEEEEEE  SSSSSSSS   SSSSSSSS
CC        HH      HH  MMMM  MMMM  MMMM  MMMM  EE               SS         SS
CC        HH      HH  MMMM  MMMM  MMMM  MMMM  EE               SS         SS
CC        HH      HH  MM  MM  MM  MM  MM  MM  EE               SS         SS
CC        HH      HH  MM  MM  MM  MM  MM  MM  EE               SS         SS
CC        HHHHHHHHHH  MM      MM  MM      MM  EEEEEEE      SSSSS      SSSSS
CC        HHHHHHHHHH  MM      MM  MM      MM  EEEEEEE      SSSSS      SSSSS
CC        HH      HH  MM      MM  MM      MM  EE                 SS         SS
CC        HH      HH  MM      MM  MM      MM  EE                 SS         SS
CC        HH      HH  MM      MM  MM      MM  EE                 SS         SS
CC        HH      HH  MM      MM  MM      MM  EE                 SS         SS
CCCCCCCC  HH      HH  MM      MM  MM      MM  EEEEEEEEEE  SSSSSSSS   SSSSSSSS    ....
CCCCCCCC  HH      HH  MM      MM  MM      MM  EEEEEEEEEE  SSSSSSSS   SSSSSSSS    ....
                                                                               ....
                                                                               ....
LL             IIIIII     SSSSSSSS
LL             IIIIII     SSSSSSSS
LL               II            SS
LL               II            SS
LL               II            SS
LL               II            SS
LL               II        SSSSSS
LL               II        SSSSSS
LL               II              SS
LL               II              SS
LL               II              SS
LL               II              SS
LLLLLLLLLL     IIIIII     SSSSSSSS
LLLLLLLLLL     IIIIII     SSSSSSSS
```

```
   1    0001   0  %TITLE 'EDT$CHMMESS - output a message'
   2    0002   0  MODULE EDT$CHMMESS (                              ! Output a message
   3    0003   0                  IDENT = 'V04-000'                 ! File: CHMMESS.BLI Edit: JBS1010
   4    0004   0                  ) =
   5    0005   1  BEGIN
   6    0006   1
   7    0007   1  !****************************************************************************
   8    0008   1  !*                                                                          *
   9    0009   1  !*    COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
  10    0010   1  !*    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
  11    0011   1  !*    ALL RIGHTS RESERVED.                                                  *
  12    0012   1  !*                                                                          *
  13    0013   1  !*    THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
  14    0014   1  !*    ONLY IN ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
  15    0015   1  !*    INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
  16    0016   1  !*    COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
  17    0017   1  !*    OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
  18    0018   1  !*    TRANSFERRED.                                                          *
  19    0019   1  !*                                                                          *
  20    0020   1  !*    THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
  21    0021   1  !*    AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
  22    0022   1  !*    CORPORATION.                                                          *
  23    0023   1  !*                                                                          *
  24    0024   1  !*    DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
  25    0025   1  !*    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.               *
  26    0026   1  !*                                                                          *
  27    0027   1  !*                                                                          *
  28    0028   1  !****************************************************************************
  29    0029   1  !
  30    0030   1
  31    0031   1  !++
  32    0032   1  ! FACILITY:      EDT -- The DEC Standard Editor
  33    0033   1  !
  34    0034   1  ! ABSTRACT:
  35    0035   1  !
  36    0036   1  !       This module outputs a message on the last line of the terminal.
  37    0037   1  !
  38    0038   1  ! ENVIRONMENT:  Runs at any access mode - AST reentrant
  39    0039   1  !
  40    0040   1  ! AUTHOR: Bob Kushlis, CREATION DATE: Unknown
  41    0041   1  !
  42    0042   1  ! MODIFIED BY:
  43    0043   1  !
  44    0044   1  ! 1-001 - Original.  DJS 04-Feb-1981.  This module was created by
  45    0045   1  !         extracting the routine EDT$$OUT_MSG  from module CHANGE.BLI.
  46    0046   1  ! 1-002 - Regularize headers.  JBS 03-Mar-1981
  47    0047   1  ! 1-003 - Make this routine general for messages or strings.  SMB 30-Jun-1982
  48    0048   1  ! 1-004 - Remove setting to TI_WRSTR in case HCPY change mode.  SMB 02-Jul-1982
  49    0049   1  ! 1-005 - Set EDT$$G_LASTMSG.  JBS 05-Jul-1982
  50    0050   1  ! 1-006 - Call EDT$$FMT_STR instead of EDT$$FMT_LIT, so EDT$$G_PRV_COL will be
  51    0051   1  !         kept up to date.  JBS 05-Oct-1982
  52    0052   1  ! 1-007 - Don't clear EDT$$G_SCR_CHGD.  JBS 09-Oct-1982
  53    0053   1  ! 1-008 - Bypadd most of the fancy stuff if we are in hardcopy change mode.  JBS 16-Nov-1982
  54    0054   1  ! 1-009 - Check for terminal type unkown also.  SMB 03-Dec-1982
  55    0055   1  ! 1-010 - Treat message number 0 as meaning no message.  JBS 01-Apr-1983
  56    0056   1  !--
  57    0057   1
```

```
  59    0058  1 %SBTTL 'Declarations'
  60    0059  1 !
  61    0060  1 ! TABLE OF CONTENTS:
  62    0061  1 !
  63    0062  1
  64    0063  1 REQUIRE 'EDTSRC:TRAROUNAM';
  65    0502  1
  66    0503  1 FORWARD ROUTINE
  67    0504  1     EDT$$OUT_MSG : NOVALUE;                          ! Output a message on the last line of the terminal
  68    0505  1 !
  69    0506  1 !
  70    0507  1 ! INCLUDE FILES:
  71    0508  1 !
  72    0509  1
  73    0510  1 REQUIRE 'EDTSRC:EDTREQ';
  74    0645  1
  75    0646  1 !
  76    0647  1 ! MACROS:
  77    0648  1 !
  78    0649  1 !     NONE
  79    0650  1 !
  80    0651  1 ! EQUATED SYMBOLS:
  81    0652  1 !
  82    0653  1 !     NONE
  83    0654  1 !
  84    0655  1 ! OWN STORAGE:
  85    0656  1 !
  86    0657  1 !     NONE
  87    0658  1 !
  88    0659  1 ! EXTERNAL REFERENCES:
  89    0660  1 !
  90    0661  1 !     In the routine
```

```
 92    0662  1  %SBTTL 'EDT$$OUT_MSG - output a message'
 93    0663  1
 94    0664  1  GLOBAL ROUTINE EDT$$OUT_MSG (                           ! Output a message
 95    0665  1      POS,                                                ! Line number for this message
 96    0666  1      MESS,                                               ! The message number to output
 97    0667  1      ADDR,                                               ! Address of a string
 98    0668  1      LEN                                                 ! Length of a string
 99    0669  1      ) : NOVALUE =
100    0670
101    0671  1  !++
102    0672  1  !  FUNCTIONAL DESCRIPTION:
103    0673  1  !
104    0674  1  !       This routine outputs a message on the last line of the terminal.
105    0675  1  !       The input parameters are the line position for the message,
106    0676  1  !       the message number (if it is a message), or the message string and
107    0677  1  !       its length if no message number is present.
108    0678  1  !
109    0679  1  !  FORMAL PARAMETERS:
110    0680  1  !
111    0681  1  !   POS                    The line number on which to print message
112    0682  1  !
113    0683  1  !   MESS                   The number of the message to output
114    0684  1  !
115    0685  1  !   ADDR                   The address of a string message
116    0686  1  !
117    0687  1  !   LEN                    The length of the string message
118    0688  1  !
119    0689  1  !  IMPLICIT INPUTS:
120    0690  1  !
121    0691  1  !       EDT$$A_FMT_WRRUT
122    0692  1  !       EDT$$G_MESSAGE_LINE
123    0693  1  !       EDT$$G_TI_TYP
124    0694  1  !
125    0695  1  !  IMPLICIT OUTPUTS:
126    0696  1  !
127    0697  1  !       EDT$$G_PRV_COL
128    0698  1  !       EDT$$G_TIN_ECHOPOS
129    0699  1  !       EDT$$G_MSGFLG
130    0700  1  !       EDT$$G_LASTMSG
131    0701  1  !
132    0702  1  !  ROUTINE VALUE:
133    0703  1  !
134    0704  1  !       NONE
135    0705  1  !
136    0706  1  !  SIDE EFFECTS:
137    0707  1  !
138    0708  1  !       NONE
139    0709  1  !
140    0710  1  !--
141    0711  1
142    0712  2      BEGIN
143    0713  2
144    0714  2      EXTERNAL ROUTINE
145    0715  2          EDT$$STOP_WKINGMSG,                             ! Stop the working message
146    0716  2          EDT$$FMT_STR,                                   ! Put a string in format buffer
147    0717  2          EDT$$OUT_FMTBUF,                                ! Output the format buffer
148    0718  2          EDT$$SC_POSCSIF,                                ! Put cursor position in format buffer
```

```
 149   0719  2              EDT$$SC_ERATOEOL,                            ! Erase to end of line
 150   0720  2              EDT$$SC_REVID,                               ! Start reverse video
 151   0721  2              EDT$$MSG_TOSTR;                              ! Get message text
 152   0722  2
 153   0723  2          EXTERNAL
 154   0724  2              EDT$$G_MESSAGE_LINE,                         ! Message line
 155   0725  2              EDT$$G_TIN_ECHOPOS,                          ! Column to start message echo
 156   0726  2              EDT$$G_MSGFLG,                               ! 1 = erase the message line on the next keystroke
 157   0727  2              EDT$$G_PRV_COL,                              ! Previous column number.
 158   0728  2              EDT$$G_LASTMSG,                              ! The last message printed
 159   0729  2              EDT$$G_TI_TYP;                               ! The type of the terminals
 160   0730
 161   0731  2          EDT$$STOP_WKINGMSG ();
 162   0732  2  !+
 163   0733  2  ! If the message is being printed on the last line, then we want it to
 164   0734  2  ! stay there until the user hits a key, but we don't want to issue the
 165   0735  2  ! PRTC message.
 166   0736  2  !-
 167   0737
 168   0738  3          IF (.POS EQL .EDT$$G_MESSAGE_LINE + 1)
 169   0739          THEN
 170   0740              BEGIN
 171   0741  3              EDT$$G_TIN_ECHOPOS = 0;
 172   0742  3              EDT$$G_MSGFLG = 1;
 173   0743  2              END;
 174   0744
 175   0745  2  !+
 176   0746  2  ! Don't do anything fancy if this is a hard copy terminal.
 177   0747  2  !-
 178   0748  2
 179   0749  3          IF (.EDT$$G_TI_TYP NEQ TERM_HCPY) AND (.EDT$$G_TI_TYP NEQ TERM_UNKNOWN)
 180   0750          THEN
 181   0751              BEGIN
 182   0752  3  !+
 183   0753  3  ! Force the cursor to the indicated line and column
 184   0754  3  !-
 185   0755  3              EDT$$SC_POSCSIF (.POS, .EDT$$G_TIN_ECHOPOS);
 186   0756  3  !+
 187   0757  3  ! Erase the line.
 188   0758  3  !-
 189   0759  3              EDT$$SC_ERATOEOL ();
 190   0760  3  !+
 191   0761  3  ! Turn reverse video on.
 192   0762  3  !-
 193   0763  3              EDT$$SC_REVID ();
 194   0764  2              END;
 195   0765
 196   0766  2  !+
 197   0767  2  ! Get the message.
 198   0768  2  !-
 199   0769
 200   0770  3          IF (.LEN NEQ 0)
 201   0771          THEN
 202   0772              BEGIN
 203   0773  3              EDT$$FMT_STR (.ADDR, .LEN);
 204   0774  3              EDT$$G_LASTMSG = 1;
 205   0775  3              END
```

```
EDTSCHMMESS       EDTSCHMMESS - output a message              L 2                                                    Page  5
V04-000           EDTS$OUT_MSG - output a message            16-Sep-1984 00:04:03   VAX-11 Bliss-32 V4.0-742
                                                             14-Sep-1984 12:22:38   DISK$VMSMASTER:[EDT.SRC]CHMMESS.BLI;1   (3)
```

```
; 206    0776   2        ELSE
; 207    0777   2
; 208    0778   3            IF (.MESS NEQ 0)
; 209    0779   3            THEN
; 210    0780   3                BEGIN
; 211    0781   3                EDTS$MSG_TOSTR (.MESS);
; 212    0782   3                EDTS$G_LASTMSG = .MESS;
; 213    0783   3                END;
; 214    0784   2
; 215    0785   2    !+
; 216    0786   2    ! Write out the buffer.
; 217    0787   2    !-
; 218    0788   2        EDTS$OUT_FMTBUF ();
; 219    0789   1        END;                                    ! of routine EDTS$OUT_MSG


                                .TITLE   EDTSCHMMESS EDTSCHMMESS - output a message
                                .IDENT   \V04-000\

                                .EXTRN   EDTS$STOP_WKINGMSG
                                .EXTRN   EDTS$FMT_STR, EDTS$OUT_FMTBUF
                                .EXTRN   EDTS$SC_POSCSIF
                                .EXTRN   EDTS$SC_ERATOEOL
                                .EXTRN   EDTS$SC_REVID, EDTS$MSG_TOSTR
                                .EXTRN   EDTS$G_MESSAGE_LINE
                                .EXTRN   EDTS$G_TIN_ECHOPOS
                                .EXTRN   EDTS$G_MSGFLG, EDTS$G_PRV_COL
                                .EXTRN   EDTS$G_LASTMSG, EDTS$G_TI_TYP

                                .PSECT   _EDTSCODE,NOWRT, SHR, PIC,2

                  001C 00000    .ENTRY   EDTS$OUT_MSG, Save R2,R3,R4          ; 0664
       54 00000000G  00   9E 00002    MOVAB    EDTS$G_LASTMSG, R4
       53 00000000G  00   9E 00009    MOVAB    EDTS$G_TIN_ECHOPOS, R3
    50 00000000G  00        00   FB 00010    CALLS    #0, EDTS$STOP_WKINGMSG   ; 0731
    50 00000000G  00        01   C1 00017    ADDL3    #1, EDTS$G_MESSAGE_LINE, R0  ; 0738
             50        04   AC   D1 0001F    CMPL     POS, R0
                           09   12 00023    BNEQ     1$
                           63   D4 00025    CLRL     EDTS$G_TIN_ECHOPOS       ; 0741
    00000000G  00          01   D0 00027    MOVL     #1, EDTS$G_MSGFLG        ; 0742
       50 00000000G  00    00   D0 0002E 1$: MOVL    EDTS$G_TI_TYP, R0        ; 0749
                   03        50   D1 00035    CMPL     R0, #3
                           1E   13 00038    BEQL     2$
                           50   D5 0003A    TSTL     R0
                           1A   13 0003C    BEQL     2$
                           63   DD 0003E    PUSHL    EDTS$G_TIN_ECHOPOS       ; 0755
                      04   AC   DD 00040    PUSHL    POS
    00000000G  00          02   FB 00043    CALLS    #2, EDTS$SC_POSCSIF
    00000000G  00          00   FB 0004A    CALLS    #0, EDTS$SC_ERATOEOL     ; 0759
    00000000G  00          00   FB 00051    CALLS    #0, EDTS$SC_REVID        ; 0763
                      10   AC   D5 00058 2$: TSTL    LEN                      ; 0770
                           10   13 0005B    BEQL     3$
                7E   0C   AC   7D 0005D    MOVQ     ADDR, -(SP)              ; 0773
    00000000G  00          02   FB 00061    CALLS    #2, EDTS$FMT_STR
                           64        01   D0 00068    MOVL     #1, EDTS$G_LASTMSG   ; 0774
                           12   11 0006B    BRB      4$                       ; 0770
             52   08   AC   D0 0006D 3$: MOVL    MESS, R2                     ; 0778
```

```
                                          0C 13 00071          BEQL     4$
                                          52 DD 00073          PUSHL    R2                                              ; 0781
                      00000000G 00     01 FB 00075          CALLS    #1, EDT$$MSG_TOSTR
                                64     52 DO 0007C          MOVL     R2, EDT$$G_LASTMSG                               ; 0782
                      00000000G 00  00 FB 0007F 4$:         CALLS    #0, EDT$$OUT_FMTBUF                              ; 0788
                                          04 00086          RET                                                       ; 0789
```

; Routine Size: 135 bytcs,   Routine Base: _EDT$CODE + 0000

```
;  220          0790  1
;  221          0791  1 !<BLF/PAGE>
```

```
EDT$CHMMESS    EDT$CHMMESS - output a message              16-Sep-1984 00:04:03   VAX-11 Bliss-32 V4.0-742         Page  7
V04-000        EDT$$OUT_MSG  - output a message            14-Sep-1984 12:22:38   DISK$VMSMASTER:[EDT.SRC]CHMMESS.BLI;1   (4)
; 223          0792 1 END                                          ! of module EDT$CHMMES
; 224          0793 1
; 225          0794 0 ELUDOM
```

```
;                        PSECT SUMMARY
;
;    Name                        Bytes                       Attributes
;
; _EDT$CODE                          135  NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
```

```
;                        Library Statistics
;
;                                   -------- Symbols --------    Pages      Processing
;    File                           Total   Loaded   Percent    Mapped     Time
;
; _$255$DUA28:[EDT.SRC]EDT.L32;1       377      2         0        40       00:00.2
; _$255$DUA28:[EDT.SRC]PSECTS.L32;1      2      1        50         7       00:00.1
```

```
;                        COMMAND QUALIFIERS
;
;    BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:CHMMESS/OBJ=OBJ$:CHMMESS MSRC$:CHMMESS.BLI/UPDATE=(ENH$:CHMMESS
;    )
```

```
; Size:          135 code + 0 data bytes
; Run Time:         00:11.9
; Elapsed Time:     00:15.0
; Lines/CPU Min:    4006
; Lexemes/CPU-Min: 10289
; Memory Used:   72 pages
; Compilation Complete
```

CHMMRKCHG
LIS

CHMPARSEN
LIS

CHMSELPOS
LIS

CHMSPLLIN
LIS

DATA
LIS

CHMPAREN
LIS

CHMUNDEL
LIS

COMMAND
LIS

CHMONSTR
LIS

CHMSAVPOS
LIS

CHMSCHSTR
LIS

CHMREPOS
LIS

CHMMESS
LIS

CHMPASTE
LIS

CHMSENDEL
LIS

CHMTADJ
LIS

CHMSAVTXT
LIS

CLRKEY
LIS

CHMNEWLEN
LIS

CHMPARSE
LIS

CHMSAVLIN
LIS

CHMSUBS
LIS